

composition of electronic collections and the set of services that determine the functionality of the digital library.

Keywords: digital library, digital mathematical library, Lobachevskii DML, text analysis, matadata.

УДК 519.688

ОБУЧАЮЩЕ-КОНТРОЛИРУЮЩИЕ ПРОГРАММЫ-ТРЕНАЖЕРЫ В СРЕДЕ MATHEMATICA

Ж.И. Зайцева¹

¹ zhanna-zji79@mail.ru; Казанский (Приволжский) федеральный университет, Набережночелнинский институт

В статье описывается работа программы-тренажера по теме «Ряды Фурье», созданной в компьютерной математической системе Mathematica, приводится и сама программа.

Ключевые слова: компьютерная программа, тренажер, Mathematica, информационные технологии.

Одной из сторон процесса информатизации высшего образования является использование информационных технологий в процессе изучения математики в вузах путем создания и использования новых форм педагогических программных продуктов с применением средств информационных технологий. Для изучения отдельных разделов и тем по математике предлагаем использовать в учебном процессе компьютерные программы-тренажеры, составленные и реализованные в системе Mathematica.

Программа-тренажер — это педагогический программный продукт, созданный и используемый в среде Mathematica, его действие основывается на специфических программных процедурах языка программирования Wolfram Language. Тренажер заключается в одной скрытой нередактируемой ячейке и представляет собой обучающе-контролирующую программу, использовать которую целесообразно как в процессе аудиторной работы, так и в домашних условиях для самостоятельной работы студентов очной, заочной и дистанционной форм обучения студентов [1, с. 21]. Работа тренажера отличается от простого тестирования тем, что в процессе решения поставленного примера вводится не только ответ, но и промежуточные вычисления.

Рассмотрим работу тренажера по теме «Ряды Фурье», когда тестируемый выполняет задание о разложении функции в ряд Фурье. В начале работы появляется запись: «Далее выделите скрытую ячейку левой кнопкой мышки, нажмите по очереди Shift+Enter и следуйте дальнейшим указаниям». После того как выполнено данное указание, на экране появляется диалоговое окно с надписью, что нужно в нем писать, и тренажер вступает в диалог с пользователем (рисунок 1).

В диалоговое окно вводится фамилия, имя, группа студента и условие примера. После введения условия примера его нужно решить и записать в диалоговое окно полученный ответ. Все производимые действия отражаются в рабочем поле. Если введенный результат правильный, то в рабочем поле появится запись «Молодец!

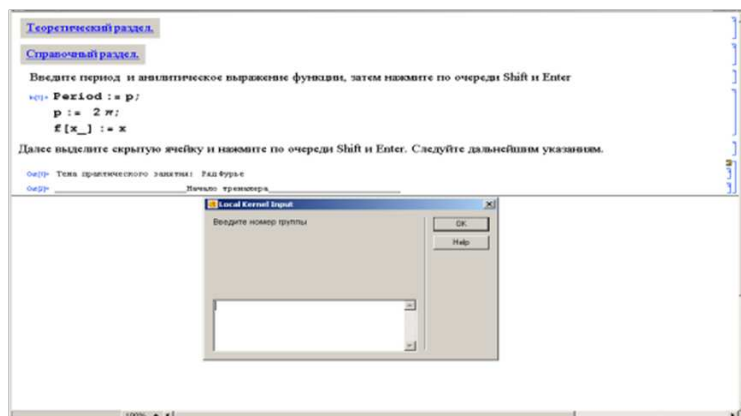


Рис. 1. Работа тренажера

Количество ошибок 0», и пользователь переходит к выполнению дальнейших указаний; в противном случае в рабочем поле появится другая запись: «Вы допустили ошибку. У вас осталось попыток: 2». В этом случае есть возможность самостоятельно найти ошибку в решении примера; для этого можно обратиться к теоретической стороне данного примера, просмотреть типовые примеры, проверить, нет ли ошибки ввода. Следует переделать промежуточное вычисление и записать в диалоговое окно исправленное значение; если оно правильное, то в рабочем поле появится запись «Молодец. Количество ошибок 1», и можно переходить к выполнению дальнейших указаний. В противном случае будет запись «Вы допустили ошибку. У вас осталось попыток: 1», и можно вновь попытаться найти ошибку и т. д. Если на каком-то этапе решения допущены три ошибки, то тренажер прекращает работу, выводит правильный ответ и количество сделанных ошибок, а также решение типового примера. По окончании работы тренажер предоставляет возможность распечатать полученные результаты для проверки и оценки преподавателем.

При прохождении тренажера, выполняя задание о разложении функции в ряд Фурье, необходимо будет определить четность, нечетность функции, вычислить и ввести коэффициенты ряда Фурье, вычислить n -сумму ряда Фурье, в соответствии с выполняемым вариантом, всего для достижения положительного результата можно совершить не более 14 ошибок [3]. Далее, если необходимо, то имеется возможность вывести полученный результат через принтер, чтобы тестирующий мог проверить соответствие предложенного и выполняемого варианта, какие ошибки совершались и какие ошибки исправлялись, т. е. в распечатке будут содержаться все действия тестируемого.

Следует отметить, что тестируемый не имеет возможности переключиться на вычислительный режим системы Mathematica до тех пор, пока не выйдет из режима тренажера.

Приведем программные процедуры работы тренажера теме «Ряды Фурье». Входные ячейки печатаем полужирным шрифтом, комментарии выделяем, как это принято в системе Mathematica, значками (* и *)

```
im = Input["Введите фамилию и имя"]; (* Встроенная функция Input вызывает
диалоговое окно; требуется ввод пользователя *)
gr = Input["Введите номер группы"];
```

```

f[x_] = Input[ "Введите функцию, разлагаемую в ряд Фурье, соответствующую
Вашему варианту"]; (* Нижнее подчеркивание дает знать системе, что x — име-
нованный аргумент *)
p = Input["Введите период функции"];
Print["Тема практического занятия: "Ряды Фурье" "];
Print["_____ Начало тренажера _____ "];
l := p/2;
Print["Студент", im, "из группы", gr];
Print["Ваше задание: разложить в ряд Фурье функцию f(x)=", f[x], "с периодом
P=2*", l];
xmin := -l;
xmax := l;
If[f[xmin] == f[-xmin] && f[xmax] == f[-xmax],
  "Четная";
  a0 :=  $\frac{2}{l} * \int_0^l f[x] dx$ ;
  ak :=  $\frac{2}{l} * \int_0^l f[x] * \text{Cos}[\frac{\pi * k * x}{l}] dx$ ;
  bk := 0;
  ChetPC := 1;
,
  If[f[xmin] == -f[-xmin] && f[xmax] == -f[-xmax],
    "Нечетная";
    a0 := 0;
    ak := 0;
    bk :=  $\frac{2}{l} * \int_0^l f[x] * \text{Sin}[\frac{\pi * k * x}{l}] dx$ ;
    ChetPC := 2;
,
    "Общего вида";
    a0 :=  $\frac{1}{l} * \int_{-l}^l f[x] dx$ ;
    ak :=  $\frac{1}{l} * \int_{-l}^l f[x] * \text{Cos}[\frac{\pi * k * x}{l}] dx$ ;
    bk :=  $\frac{1}{l} * \int_{-l}^l f[x] * \text{Sin}[\frac{\pi * k * x}{l}] dx$ ;
    ChetPC := 3;
  ];
];
Print["Определите четность функции."];
Print["Если функция четная, введите 1"];
Print["Если функция нечетная, введите 2"];
Print["Если функция общего вида, введите 3"];
i := 1; j := 3; os = 0;
While[i < j,

```

```

i += 1;
Chet = Input["Четная 1, нечетная 2, общего вида 3"];
If[Chet === ChetPC,
  Print["Молодец"];
  Break[ ]; (* Прервать цикл *)
,
  Print["У Вас осталось попыток: j - i];
  os += 1
]
];
i := 1; j := 4; ol = 0;
While[i < j, (* Цикл *)
  i += 1;
  Print["Вычислите и введите значение коэффициента a0."];
  integ1 = Input["Вычислите и введите значение коэффициента a0."];
  If[ a0 === integ1,
    Print["Вами введено значение коэффициента a0 правильно, продол-
жайте работу"];
    Break[ ]; (* Прервать цикл *)
  ,
    Print["Вы ввели a0= ", integ1 , "неправильно. У Вас осталось попыток:",
      j - i];
    ol += 1
  ];
  If[ol === 2,
    Print["Обратитесь к теоретическому разделу или к справочному"]
  ]
];
i := 1; j := 4; ok = 0;
While[i < j,
  i += 1;
  Print["Вычислите и введите значение коэффициента ak."];
  integ2 = Input["Вычислите и введите значение коэффициента ak."];
  If[ ak === integ2,
    Print["Вами введено значение коэффициента ak правильно, продол-
жайте работу"];
    Break[ ];
  ,
    Print["Вы ввели ak=", integ2 , "неправильно. У Вас осталось попыток:",
      j - i];
    ok += 1
  ]
];
If[ok === 2,
  Print["Обратитесь к теоретическому разделу или к справочному"]
]

```

```

];
i := 1; j := 4; oc = 0;
While[i < j,
  i += 1;
  Print["Вычислите и введите значение коэффициента bk."];
  integ3 = Input["Вычислите и введите значение коэффициента bk."];
  If[ bk === integ3,
    Print["Вами введено значение коэффициента bk правильно, продол-
жайте          работу"];
    Break[ ];
  ,
    Print["Вы ввели bk=", integ3, "неправильно. У Вас осталось попыток:",
      j - i];
    oc += 1
  ];
  If[oc === 2,
    Print["Обратитесь к теоретическому разделу или к справочному"]
  ]
];
Print["Коэффициенты ряда Фурье имеют вид:"]
Print["a0= a0"];
Print["ak= ak"];
Print["bk= bk"];
n = Input[
  "Введите значение порядка n частичных сумм ряда Фурье для аппрокси-
мации функции, соответствующее вашему варианту"];

$$f1[x] := \frac{a0}{2} + \sum_{k=1}^n \left( ak * \cos\left[\frac{\pi * k * x}{1}\right] + bk * \sin\left[\frac{\pi * k * x}{1}\right] \right);$$

i := 1; j := 4; op = 0;
While[i < j,
  i += 1;
  Print["Вычислите и введите n=", n, "частичных сумм ряда Фурье"];
  s = Input["Вычислите и введите значение частичных сумм ряда Фурье"];
  If[ s === f1[x],
    Print["Вами введено выражение частичных сумм ряда Фурье правильно"];
    Break[ ];
  ,
    Print["Вы ввели bk=", s, "неправильно. У Вас осталось попыток:", j - i];
    op += 1
  ];
  If[op === 2,
    Print["Обратитесь к теоретическому разделу или к справочному"]
  ]
];
Print["Ряд Фурье функции f(x)=", f[x], "имеет вид при n=", n , f(x)", f1[x]];

```

```

s := os + ol + ok + oc + op;
Print["Вы ошиблись при прохождении тренажера", s, "раз"];
Print["_____ Конец тренажера _____"];
If[0 <= s <= 2,
  Print["Студент", im, "группы", gr, "тренажер выполнил на оценку отлич-
но"]
];
If[3 <= s <= 5,
  Print["Студент", im, "группы", gr, "тренажер выполнил на оценку хорошо"]
];
If[6 <= s <= 8,
  Print["Студент", im, "группы", gr, "тренажер выполнил на оценку
удовлетворительно"]
];
If[9 <= s <= 14,
  Print["Студент", im, "группы", gr, "тренажер выполнил на оценку
неудовлетворительно"]
]

```

Тело тренажера заключается в одной скрытой ячейке. Программа его такова, что компьютер параллельно с тестируемым ведёт вычисления и сравнивает свой результат с введенным, давая тестируемому положительный или отрицательный отзыв. Поэтому подобных однотипных примеров можно проходить на тренажере любое количество и это очень важно, так как программа не меняется, а меняется условие задачи (то есть вводимая функция).

Можно также, используя среду Mathematica, построить график функции, что нужно для определения суммы ряда Фурье на концах отрезка и в точках разрыва функции внутри интервала; при этом используется теорема Дирихле. Так как мы создавали единую программу-тренажер, то в самом тренажере предусмотреть столь тонкие математические нюансы оказалось довольно сложно, но стоит работать над этим. Для использования приведенного выше тренажера подбираются только функции, заданные одним аналитическим выражением и имеющие разрыв лишь на концах отрезка.

В программе-тренажере находится n -частичная сумма ряда Фурье, на экран можно вывести ее график совместно с графиком самой функции, что наглядно продемонстрирует аппроксимацию функции рядом Фурье.

Отметим, что тренажер по тому же принципу (используя ту же основу программы) можно составить не только на одну тему, но и для изучения различных тем и разделов математики, где требуется численная характеристика искомых величин, например, «Матрицы и действия над ними», «Вычисление пределов функции», «Дифференциальное исчисление функций одной переменной», «Дифференциальное исчисление функций нескольких переменных», «Вычисление определенных интегралов», «Вычисление площадей, объемов, длин дуг», «Ряды Фурье» и т.д.

Использование программы-тренажера в образовательном процессе не заменяет традиционные формы преподавания, но дополняет и обогащает их. Грамотное составление и использование поможет существенно интенсифицировать учебный

процесс, сделать процесс обучения математике более привлекательным и интересным и подготовить студента к квалифицированному применению компьютера.

Литература

1. Зайниев Р.М. Применение компьютерных технологий в математическом образовании студентов технических направлений подготовки / Р.М. Зайниев, Ж.И. Зайцева // Высшее образование сегодня. – 2015. – № 1. – С. 19–22.
2. Зайцева Ж.И. Методика преподавания высшей математики с применением новых информационных технологий (в техническом вузе): автореф. дисс ... канд.пед.наук. – М., 2005. – 27 с.
3. Программа тренажера для решения задач по математике, созданного в компьютерной математической системе Mathematica, по теме «Ряды Фурье»: свидетельство о государственной регистрации программы для ЭВМ № 2014661421 / Ж.И. Зайцева. – № 2014661421; 2014.

THE TEACHING-MONITORING SIMULATOR PROGRAMS IN THE ENVIRONMENT MATHEMATICA

Zh.I. Zaytceva

This paper describes the work of the simulator program on the topic “Fourier Series” created in the computer mathematical system Mathematica, the program is shown as well.

Keywords: computer program, simulator, Mathematica, information technology.

УДК 004, 519.852

ПРИМЕНЕНИЕ СКМ MAPLE ПРИ РЕШЕНИИ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Н.В. Зайцева¹

¹ n.v.zaiceva@yandex.ru; Казанский (Приволжский) федеральный университет

В работе приведены возможности одной из библиотек системы компьютерной математики Maple для решения задач линейного программирования.

Ключевые слова: линейное программирование, транспортная задача, программа Maple.

Одной из задач линейного программирования является транспортная задача, суть которой состоит в составлении плана перевозок некоторой продукции от поставщиков к потребителям таким образом, чтобы суммарные потребности потребителей были полностью удовлетворены за счет вывоза запаса продукции от поставщиков. Целью транспортной задачи является минимизация суммарной стоимости всех перевозок.

Различают два вида транспортных задач: открытые – если запас продукции от поставщиков не совпадает с суммарной потребностью в продукции у потребителей, и закрытые – в случае совпадения. Следует отметить, что закрытая транспортная задача разрешима всегда.

Приведем шаги алгоритма решения транспортной задачи:

Шаг 1. Построение начального плана перевозок.